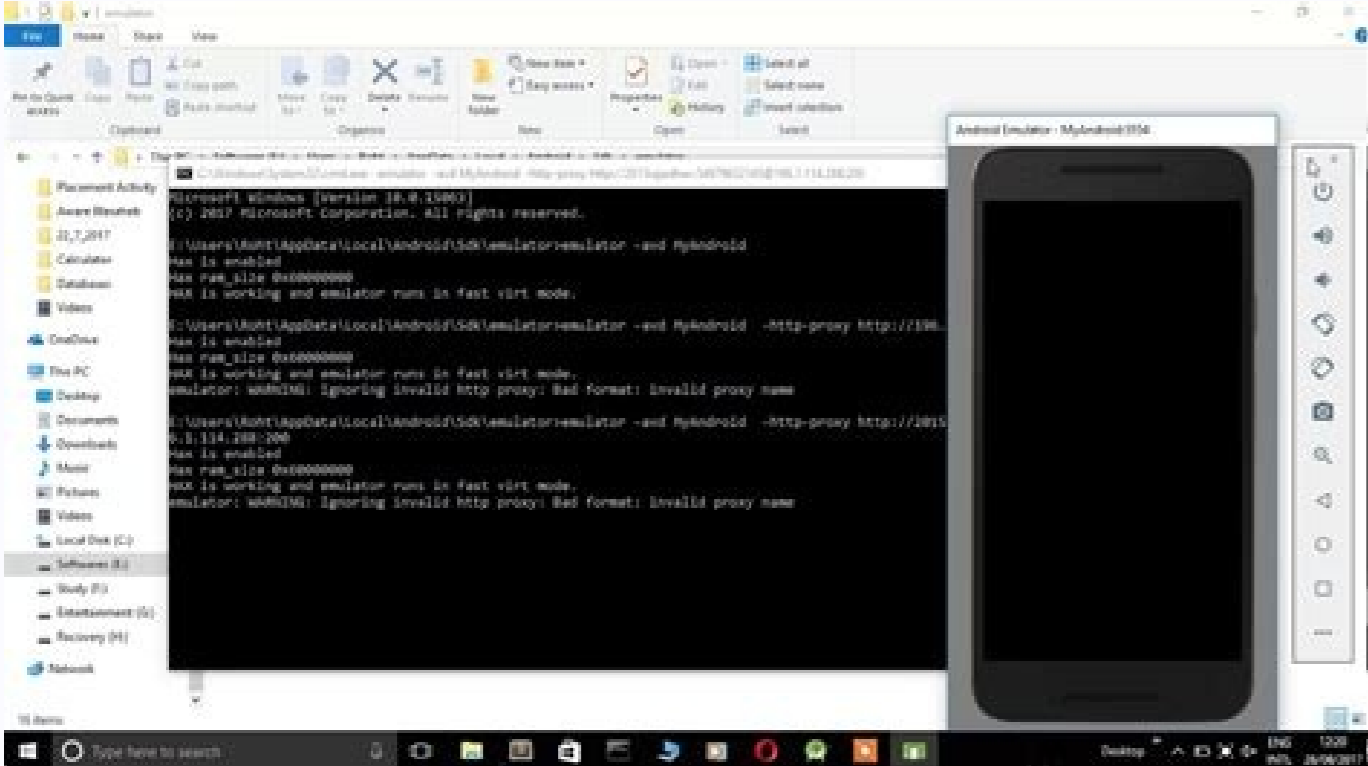
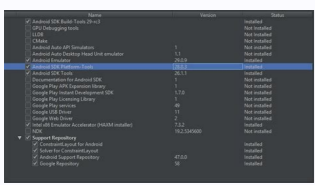
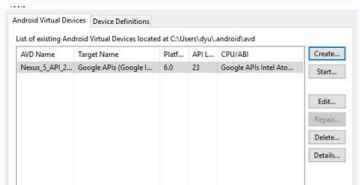
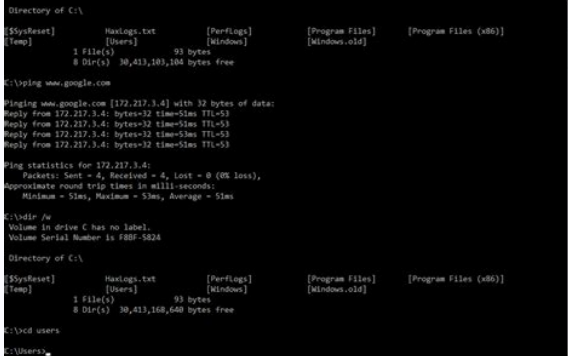


Run android emulator from command line windows

Continue



How to run android emulator with android studio. How to run android emulator on windows. How to start android emulator from command line windows 10.

The purpose of this article is to help a system administrator set up Android command-line tools on a Windows PC so that anyone can use the Android emulator and other command-line tools without installing the full Android Studio SDK. This saves disk space on the system if the user only needs the cmdline tools and doesn't need the full Android Studio development environment. Download and install the Java JDK from this link: On the download page, scroll down and click the next download link to the Windows x64 installer. A pop-up window will open with a checkbox and a download button. You have to check the box but don't directly click on the download button here as it will redirect you to the login page. Instead, right-click the button and select Copy Link from the context menu. Open a text editor, paste the copied link and change the download link as shown in the image below. Just highlight the URL in red, then change the text "otn" to "otn-pub", then paste the edited link (highlighted in green) into your browser's address bar. This will start the download. Please note that commercial use of the Oracle JDK may require the purchase of a valid license. If you don't want to buy a license, you can use OpenJDK as described in this article. Once the download is complete, install the JDK in the default location. Download and install the Android Command Line Tools Open this link in your browser and click the Download Options button: Scroll down to the Command Line Tools only section and click the download link next to Windows. This will download the zip file. The latest version of cmdline-tools has some issues, and it's rarely documented how to fix these issues. Follow these steps exactly as shown: Open the Run dialog box (press WindowsKey + R), type the following path and press Enter: C:\Users\Sreender\AppData\Local\Android\Sdk\cmdline-tools\latest Make sure you have changed your username. Now create a new folder called android and inside it another folder called cmdline-tools and inside it another folder called tools. Finally you have: C:\Users\Sreender\AppData\Local\Android\cmdline-tools\tools Now open the zip file you downloaded in step 2 with 7-Zip. Many users reported problems while using the built-in Windows Zip utility to extract the downloaded ZIP file. So we will use 7-zip (a free tool) to extract the zip file. You can download the 7-zip tool from . In the 7-zip manager you will see a folder called cmdline-tools. Double-click the cmdline-tools folder to enter it. Now you will see 4 items (bin, lib, NOTICE.txt and source.properties). Click "Extract" button and paste the path "C:\Users\Sreender\AppData\Local\Android\cmdline-tools\tools" into the "Copy to" box and click "OK". Again, be sure to change your username. Open the directory to which you extracted the files. Now rename the folder named "bin" to "newest". At the end you will get a directory structure as shown in the following image: C:\Users\Sreender\AppData\Local\Android\cmdline-tools\tools\latest You can copy this path as you will need it in the next section to configure environment variables. Configuring environment variables Open the Run dialog box (press Windows logo key + R). Type "sysdm.cpl" without quotes to open the advanced system properties. Click Environment Variables. Under User Variables, select Path and click Edit. Now click New and paste the path you captured in step 10 of the previous section. C:\Users\Sreender\AppData\Local\Android\cmdline-tools\tools\latest Repeat step 5 again to add the environment variable to the preinstalled Java JDK. FYI, on my computer JDK was installed in C:\Program Files\Java\jdk1.8.0_301\bin so I will copy this path. Now click on OK and you can proceed to the next section. Updating and starting the SDK Manager If everything you've done so far is correct, you shouldn't have any problems. Follow these steps to update the SDK Manager command line tool: Open the Run dialog box, type cmd, hold down Ctrl + Shift and press Enter. Click YES in the popup. An elevated command prompt will open. Now type "sdkmanager" without the quotes and press Enter. If you don't get any errors, it's successful. General Problems Problem 1 You receive an error message stating that "sdkmanager is not recognized as an internal or external command, executable or batch file". Solutions. Go to the "Set Environment Variables" section and make sure you're correct. Added environment variable and copied correct path to Android cmdline-tools. Watch this video to learn how to set environment variables in Windows. Problem 2: I get the error message "Error: The main class could not be found or loaded." Solutions: You must ensure that you extract the android-cmdline-tools to your user profile in your AppData directory. If you extracted it to the Program Files directory, you may get this error. Another thing you should do is go back to the Set Environment Variables section and make sure the environment variables are added correctly. Issue 3 You get a warning, "Warning: Could not load source lists!" OR "Warning: I/O exception while loading manifest!" OR "Warning: Waiting for package manifest to load remote" Solutions: You need to check, if something is blocking the connection. Most likely it will be your antivirus software. You can disable it temporarily, or at least disable scanning for encrypted connections. Installing platform tools, system images and APIs Now we can install platform tools, system images, Google API layers, etc. directly from the command line To do this, do the following: Open the runtime dialog, type cmd and hold Ctrl+Shift+press Enter. Press YES in the pop-up window. An elevated command prompt will open. Type the following command and press Enter: sdkmanager "platform-tools" "platforms;android-29" "system-images;android-29;google apis;x86_64" In the command above, "android-29" means Android 10 version. If you want download and try android 11, you can replace "android-29" with "android-30". You can install multiple versions of Android at the same time, but make sure you have enough disk space. A license agreement will be offered. Type Y to agree and press Enter to continue. This command will take some time as it downloads many packets from the internet. Make sure you have an unmetered connection that doesn't charge for data. After executing the above command, several directories will be created in C:\Users\Sreender\AppData\Local\Android. Then we add the environment variable to the emulator directory. Go to the emulator directory, click on the address bar of File Explorer and copy the path. Repeat steps 1 through 5 in Setting environment variables to add the emulator directory to the path. Now we will create our first virtual Android device (aka AVD). To create an AVD, type the following command and press Enter: avdmanager create avd -n AVDv10 -k "system-images;android-29;google apis;x86_64" You can name the AVD by specifying -n. I chose AVDv10 because it will be an Android 10 device. You can use the avdmanager list avd command to view the created AVD files. Launching an Android Virtual Device (AVD) To boot your first Android Virtual Device, do the following: Open the run dialog, type cmd, hold down Ctrl + Shift and press Enter. An elevated command prompt will open. Type this command and press Enter: @AVDv10 emulator This command will start showing output and you will see your first running AVD in a new window. If you see an error message saying that the emulator is not recognized as an internal or external command, operating program, or batch file, make sure you have correctly added the emulator directory to the Path variable (see steps 5 and 6 in Installing Platform Tools, System Images, and APIs). Delete the Android Virtual Device (AVD) To delete the AVD you just created, you can use the following command: avdmanager delete avd -n AVDv10 Using the Android Debug Bridge (ADB) The Android Debug Bridge (adb) is a command line tool that enables communication with the Android device. The adb command facilitates various device actions like installing and debugging apps, transferring data to/from Android devices, running shell commands on Android devices, etc. If you followed this guide from the beginning, you have the adb tool already installed. Just add it to an environment variable to run it from anywhere on the command line. To add an environment variable, do the following: Open the following path in File Explorer: C:\Users\Sreender\AppData\Local\Android\platform-tools Make sure you changed your username. If you cannot find the platform-tools directory on your computer, run the following command in an elevated cmd prompt: sdkmanager "platform-tools" Copy the path "C:\Users\Sreender\AppData\Local\Android" to the "Platform-Tools" directory from the Explorer address bar. Paste the copied path to the environment variable. See steps 1 through 5 in Setting Environment Variables for more information. Once the environment variable is set, open a command prompt, type "adb version" and press Enter. If you don't see an error, you should continue to the next step. If you get the error message "adb is not recognized as an internal or external command, operating program, or batch file", verify that you have correctly followed steps 1 through 5 of this section. To display a list of connected devices, you can use the "adb devices" without quotes. To install an app using an APK file, you can useinstall apk name without quotes. To launch a shell on a connected Android device or AVD, you can use the "adb shell" command without quotes. You can use "adb push source destination" to push the file to your connected android device or AVD. You can use "adb pull source destination" to download a file from your connected android device or AVD to your computer. To shut down a connected Android device or AVD from your computer, you can use the command "adb shell reboot -p" (-p stands for -poweroff). This is just a small glimpse of what adb can do. There's a lot more to it, but you get the point. If you like this post, please share it with your family and friends. Friends.

